Optimization of Dynamic Power Consumption in Multi-Tier Gate-Level Monolithic 3D ICs

Sheng-En David Lin, Partha Pratim Pande, and Dae Hyun Kim
School of Electrical Engineering and Computer Science, Washington State University, WA, USA
Email: {slin3, pande, daehyun}@eecs.wsu.edu

***Abstract*— Monolithic three-dimensional (3D) integration enables the most fine-grained integration of transistors by stacking very thin layers and fabricating monolithic inter-layer vias as small as local vias. Thus, monolithic 3D integration is expected to provide a higher degree of wirelength reduction, performance improvement, and power saving. Due to the prospective properties of the monolithic 3D integration technology, research on *multi-layer* monolithic 3D integration that stacks more than two device layers is also ongoing. In this paper, we propose an algorithm that optimizes dynamic power consumption of gate-level monolithic 3D ICs. Under the same timing constraints, our algorithm reduces dynamic power consumption more effectively than a uniform-scaling-based placement algorithm. We also design multi-tier monolithic 3D ICs and show that our algorithm outperforms the uniform-scaling-based placement algorithm by 11.4% on average.**

I. Introduction

Three-dimensional (3D) integration provides many benefits such as shorter wirelength, higher performance, smaller footprint area, and much higher inter-tier bandwidth than traditional two-dimensional integrated circuits (2D ICs). Among various 3D integration technologies, monolithic 3D integration technology is expected to provide the highest device density [1]. In addition, the size of a monolithic inter-layer via (MIV) used to electrically connect devices in different tiers is comparable to that of a local via as shown in Figure 1, so the parasitic capacitance of an MIV is almost negligible, whereas a through-silicon via (TSV) has non-negligible area and capacitance [2], [3]. Thus, monolithic 3D integration is expected to enable the highest degree of wirelength reduction, performance improvement, and power reduction.

A simple, but effective way to place cells in 3D is to use an existing 2D placement tool to generate a high-quality 2D placement result, downscale the cell locations by a constant scaling ratio, and remove cell overlaps by partitioning [4]. This methodology, namely *a uniform-scaling-based placement algorithm*, scales down the length of each net by almost the same ratio as the constant scaling ratio, thereby reducing wirelength and dynamic power consumption. However, if the timing constraint such as the target clock frequency does not change, we can reduce the dynamic power consumption further. In this paper, we propose a detailed placement algorithm, namely a *non-uniform-scaling-based placement algorithm*, to optimize dynamic power consumption more effectively than the uniform-scaling-based 3D placement algorithm.

All the monolithic 3D IC papers presented in the literature focused on the design of two-tier 3D ICs. In fact, multi-tier (more than two tiers) 3D ICs are expected to provide more benefits than two-tier 3D ICs [5] and some monolithic 3D integration technologies can fabricate multiple device layers [6]. In this paper, therefore, we also design multi-tier monolithic 3D ICs using the uniform-scaling-based and our placement algorithms and compare the quality of the algorithms.

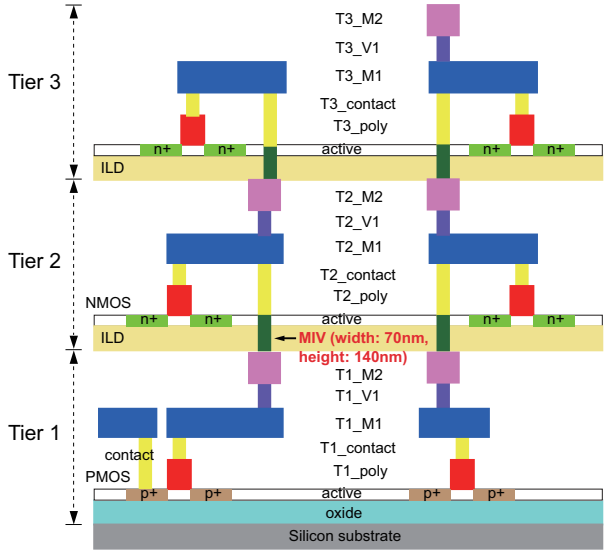

Fig. 1. Multi-tier monolithic 3D integration.

Our contributions in this paper are as follows:

- We develop a detailed placement algorithm that optimizes dynamic power consumption of monolithic 3D ICs more effectively than the uniform-scaling-based placement.
- We present theoretical background on the minimization of dynamic power consumption in monolithic 3D ICs.
- We apply the uniform-scaling-based and our placement algorithms to multi-tier monolithic 3D IC design and present their results with detailed analyses.

II. Preliminaries

In this section, we briefly review three monolithic 3D IC design methodologies presented in the literature and detail the scaling-based 3D placement algorithm.

A. Design Methodologies for Monolithic 3D ICs

Monolithic 3D ICs can be designed in several different design levels. The most fine-grained design style is the transistor-level monolithic integration (TMI) proposed in [7]. In TMI,

NMOS and PMOS transistors of each standard cell are placed in different tiers, e.g., the NMOS and PMOS transistors are placed in the top and bottom tiers, respectively. In this case, MIVs are used for both intra- and inter-cell 3D connections. TMI reduces the footprint area of each standard cell almost by half, but it overuses MIVs for inter-cell 3D routing, which increases routing complexity and leads to unroutable designs. Block-level monolithic integration (BMI) proposed in [8] is another monolithic 3D IC design methodology in which each 2D functional block is designed with 2D standard cells and all the blocks are placed in 3D using a 3D floorplanner. Thus, NMOS and PMOS transistors are placed in both bottom and top tiers and MIVs are inserted into whitespace between the blocks. Gate-level monolithic integration (GMI) proposed in [7] places 2D standard cells in 3D. GMI can reuse existing 2D standard cells and timing/power libraries. In addition, [4] proposed a design methodology using 2D placement tools for the design of gate-level monolithic 3D ICs and achieved almost 20% wirelength reduction and 16% power reduction. Thus, GMI is a prospective design methodology with respect to the design effort and the quality (wirelength, timing, and power) of 3D ICs.

B. Uniform-Scaling-Based 3D Global Placement

The 3D global placement algorithm presented in [4] works as follows. First, they determine a downscaling ratio s based on the ratio between the width (w_{2D}) of the 2D layout and the width (w_{3D}) of a target 3D layout of the design ($s = w_{\mathrm{3D}}/w_{\mathrm{2D}}$).[1] Then, they shrink the size of each cell in a given standard cell library by the scaling ratio s and place the cells in 2D using a commercial tool. By changing the library set from the downscaled one to the original one after the placement, the authors obtain a layout in which the cells overlap with each other. The overlaps are removed by partitioning, which also automatically converts the 2D layout into a 3D layout. The whole process of the downscaling of the cell size, placing cells in 2D, and restoring the original cell size is exactly the same as placing the cells first and then scaling the locations of the cells uniformly by the downscaling ratio s. Thus, we call this approach *uniform-scaling-based placement (**USBP**)*.

USBP reduces the length of each net almost by the downscaling ratio s, so the dynamic power consumption and the delay of each net are also reduced. However, the reduced net delay cannot be directly translated into higher clock frequency if the delay of the critical path is primarily due to gate delay and pin capacitance. Thus, USBP can easily reduce the dynamic power consumption, but cannot guarantee that it can increase the clock frequency. However, we can convert the increased timing margin into further dynamic power consumption. In this paper, therefore, we propose an algorithm to convert the reduced net delay in non-critical paths into power reduction by a detailed placement algorithm, which we call *non-uniform-scaling-based placement (**NUSBP**)*.

[1] Assuming both the 2D and 3D layouts have the same total silicon area, w_{3D} is $w_{\mathrm{2D}}/\sqrt{N_{\mathrm{T}}}$.

TABLE I

VARIABLES USED IN THIS PAPER

Variable	Meaning
N_{T}	# tiers
α_i	Switching activity of net i
f_{clk}	Clock frequency
V_{DD}	Supply voltage
R_i	Output resistance of cell i
$R_{\mathrm{w},i}$	Wire resistance of net i
$C_{\mathrm{w},i}$	Wire capacitance of net i
$C_{\mathrm{p},i}$	Sum of the capacitance of all input pins connected to net i

TABLE II

IDEAL BENEFITS OBTAINED BY MONOLITHIC 3D INTEGRATION AND UNIFORM-SCALING-BASED PLACEMENT.

	2D	3D
Wirelength	l	$\frac{l}{\sqrt{N_{\mathrm{T}}}}$
Wire R and C	R_w, C_w	$\frac{R_w}{\sqrt{N_{\mathrm{T}}}}$, $\frac{C_w}{\sqrt{N_{\mathrm{T}}}}$
RC delay	$\propto l^2$	$\propto \frac{l^2}{N_{\mathrm{T}}}$
Net switching power	$\propto (C_{\mathrm{w},i} + C_{\mathrm{p},i})$	$\propto (\frac{C_{\mathrm{w},i}}{\sqrt{N_{\mathrm{T}}}} + C_{\mathrm{p},i})$

III. Dynamic Power Consumption in Gate-Level Monolithic 3D ICs

In this section, we analyze dynamic power consumption in monolithic 3D ICs and investigate how we can reduce dynamic power consumption further. Table I shows the variables used in this paper and their meanings.

A. Power Reduction by Uniform Scaling

Dynamic power consumption is estimated by the following, well-known formula:

$$P_{\mathrm{int}} = \sum_{i \in N} \alpha_i \cdot f_{\mathrm{clk}} \cdot (C_{\mathrm{w},i} + C_{\mathrm{p},i}) \cdot {V_{\mathrm{DD}}}^2 \qquad (1)$$

where N is a set of all the nets in the design and we are breaking down the capacitance into two capacitive components, wire capacitance and input pin capacitance of each net. Assuming the 2D layout and the target 3D layout have the same total silicon area, the scaling factor that the USBP algorithm uses becomes $1/\sqrt{N_{\mathrm{T}}}$. Thus, the USBP algorithm ideally reduces the length of each wire by $1/\sqrt{N_{\mathrm{T}}}$, which is translated into delay and power reduction. Table II shows ideal benefits we can obtain by USBP.

Since the switching activity of each net, clock frequency, and the supply voltage are constants, we can reduce the dynamic power consumption by reducing the wire capacitance and/or the input pin capacitance as shown in Equation (1). Reducing wire capacitance requires wirelength reduction, routing layer reassignment, wire spreading, and so on. Reducing input pin capacitance requires gate sizing (downsizing in most cases).

B. Translation of Delay Benefit into Power Reduction

As shown in Table II, the USBP algorithm reduces both net delay and dynamic power consumption by wirelength reduction. As explained in Section II-B, however, increasing the clock frequency in monolithic 3D ICs is not possible

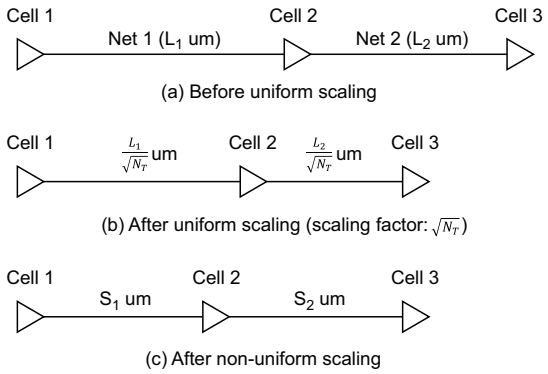

Fig. 2. Uniform- and non-uniform-scaling-based placement.

or desirable. In this case, we can adjust the cell locations to translate the delay benefit into further power reduction as shown below.

Figure 2 shows an example in which three cells are connected through two nets. Assuming that the switching activities of Net 1 and Net 2 in the figure are α_1 and α_2, respectively, the difference between the power consumptions before and after uniform scaling is:

$$\Delta P = f_{\mathrm{clk}} \cdot {V_{\mathrm{DD}}}^2 \cdot (\alpha_1 \cdot C_{\mathrm{w},1} + \alpha_2 \cdot C_{\mathrm{w},2}) \cdot (1 - \frac{1}{\sqrt{N_{\mathrm{T}}}})$$

which is the power benefit obtainable from USBP. However, we can reduce the power consumption further by moving the cells. For instance, if α_1 is greater than α_2, moving Cell 2 closer to Cell 1 along Net 1 will reduce the power consumption. Suppose Cell 2 is moved toward Cell 1 by Δx(um) after the uniform scaling. Then, the power benefit becomes

$$\begin{aligned}\Delta P = f_{\mathrm{clk}} \cdot {V_{\mathrm{DD}}}^2 \cdot (\alpha_1 \cdot C_{\mathrm{w},1} + \alpha_2 \cdot C_{\mathrm{w},2}) \cdot (1 - \frac{1}{\sqrt{N_{\mathrm{T}}}}) \\ + f_{\mathrm{clk}} \cdot {V_{\mathrm{DD}}}^2 \cdot c_{\mathrm{u}} \cdot \Delta x \cdot (\alpha_1 - \alpha_2) \qquad (2)\end{aligned}$$

where c_{u} is the capacitance per micro-meter for the nets. The second term in Equation (2) is positive because we assume that α_1 is greater than α_2. Thus, the power benefit goes up further by moving Cell 2 closer to Cell 1 in this case.

This *post-scaling adjustment of cell locations* can be performed by 1) scaling the cell locations with different scaling ratios or 2) uniformly scaling the cell locations with a constant scaling ratio ($1/\sqrt{N_{\mathrm{T}}}$) and adjusting the cell locations after the uniform scaling or 3) adjusting the cell locations before the uniform scaling and uniformly scaling the modified cell locations with a constant scaling ratio ($1/\sqrt{N_{\mathrm{T}}}$). Although all of them produce the same result, we use the third approach in this paper, but we call it *non-uniform-scaling-based placement (**NUSBP**)* as mentioned in Section II-B.

Although NUSBP reduces the power consumption further, we should take two important constraints, timing and density constraints, into account during NUSBP. The next section shows how we take the timing constraint into account and Section IV explains how we handle the density constraint.

C. Ideal Non-Uniform Scaling Under Timing Constraints

In Figure 2, suppose $d_{1,3}$ and ${d_{1,3}}'$ be the Elmore delays from the output of Cell 1 to the input of Cell 3 before and after uniform scaling, respectively. Uniform scaling of the cell locations multiplies the x- and y-coordinates of each cell by a constant scaling factor ($1/\sqrt{N_{\mathrm{T}}}$). Thus, the difference between $d_{1,3}$ and ${d_{1,3}}'$ is

$$\begin{aligned}\Delta d_{1,3} \quad &= (R_1 C_{\mathrm{w},1} + R_{\mathrm{w},1} C_{\mathrm{p},1})(1 - \tfrac{1}{\sqrt{N_{\mathrm{T}}}}) \\ &+ (R_2 C_{\mathrm{w},2} + R_{\mathrm{w},2} C_{\mathrm{p},2})(1 - \tfrac{1}{\sqrt{N_{\mathrm{T}}}}) \\ &+ \tfrac{R_{\mathrm{w},1} C_{\mathrm{w},1} + R_{\mathrm{w},2} C_{\mathrm{w},2}}{2}(1 - \tfrac{1}{N_{\mathrm{T}}}) \ .\end{aligned}$$

Since $\Delta d_{1,3}$ in the above equation is greater than zero, we can move Cell 2 along the net to reduce power consumption while still satisfying the timing constraint.

Assume that the distance between Cell 1 and Cell 2 in Figure 2 becomes S_1(um) and that between Cell 2 and Cell 3 becomes S_2(um) after non-uniform scaling. If the delay from the output of Cell 1 to the input of Cell 3 in this case is ${d_{1,3}}''$, the difference between $d_{1,3}$ and ${d_{1,3}}''$ becomes

$$\begin{aligned}\Delta {d_{1,3}}' = (R_1 C_{\mathrm{w},1} + R_{\mathrm{w},1} C_{\mathrm{p},1})(1 - \frac{S_1}{L_1}) + \frac{R_{\mathrm{w},1} C_{\mathrm{w},1}}{2}(1 - \frac{{S_1}^2}{{L_1}^2}) \\ + (R_2 C_{\mathrm{w},2} + R_{\mathrm{w},2} C_{\mathrm{p},2})(1 - \frac{S_2}{L_2}) + \frac{R_{\mathrm{w},2} C_{\mathrm{w},2}}{2}(1 - \frac{{S_2}^2}{{L_2}^2}) \ . \qquad (3)\end{aligned}$$

Setting $\Delta {d_{1,3}}'$ to zero and solving it with a constraint $S_1 + S_2 = \frac{L_1 + L_2}{\sqrt{N_{\mathrm{T}}}}$ gives us the ranges of S_1 and S_2 that do not degrade the delay of each net.

D. Practical Non-Uniform Scaling Under Delay Constraints

The ideal non-uniform scaling has several issues. Above all, it considers not a path delay but a net delay. In fact, violating timing constraints between two cells (Cell 1 and Cell 3 in Figure 2) is allowed as long as all the paths including the path between the two cells satisfy the timing constraints. However, finding timing-violation paths requires more accurate delay calculation and timing analysis engines and needs non-negligible runtime. Another issue is that the problem becomes more complex for multi-pin nets. Thus, *we assume that the delay of each net is not degraded if the length of the net after NUSBP is not longer than the length of the net before NUSBP*. However, if a net is sufficiently short, extending the net a bit (e.g., from 5um to 50um) does not significantly change the net delay in most cases. In other words, practically we can use the following inequality to preserve the delay of each net:

$$\mathrm{HPWL}_i{}' \leq \mathrm{HPWL}_i + \delta_i \qquad (4)$$

where HPWL_i is the half-perimeter wirelength of net i before non-uniform scaling, $\mathrm{HPWL}_i{}'$ is the HPWL of net i after non-uniform scaling, and δ_i is a relaxation factor, which we use to allow some delay margin for each net.

The new HPWL after non-uniform scaling is:

$$\mathrm{HPWL}_i{}' = \frac{\mathrm{HPWL}_i + \Delta\mathrm{HPWL}_i}{\sqrt{N_{\mathrm{T}}}} \qquad (5)$$

so substitution of Equation (5) into Inequality (4) gives

$$\Delta\mathrm{HPWL}_i \leq (\sqrt{N_{\mathrm{T}}} - 1)\mathrm{HPWL}_i + \sqrt{N_{\mathrm{T}}}\delta_i \qquad (6)$$

which we use to restrict the locations of the cells connected to each net to satisfy the delay constraint.

IV. Dynamic Power Optimization Algorithms

In this section, we present our algorithms in the detailed placement step for the minimization of dynamic power consumption in monolithic 3D ICs.

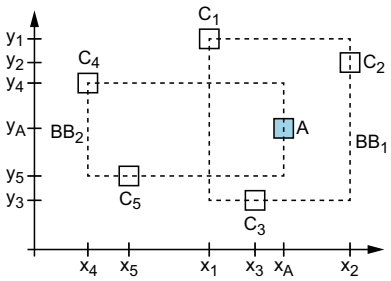

Fig. 3. Two nets and their bounding boxes. Net 1 = {A, C_1, C_2, C_3}. Net 2 = {A, C_4, C_5}.

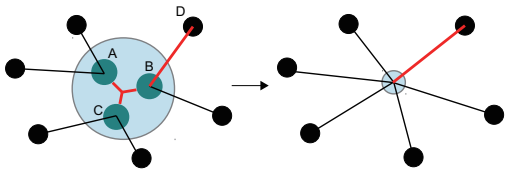

Fig. 4. Illustration of our clustering technique. The red nets are high-activity nets.

A. Overall Algorithm

For a given 2D placement result, our NUSBP algorithm modifies the cell locations in the layout and then uniformly scales the locations by $1/\sqrt{N_T}$ to generate a N_T-tier monolithic 3D IC layout. The objective is to minimize the dynamic power consumption estimated by the following formula:

$$P = f_{\text{clk}} \cdot {V_{\text{DD}}}^2 \sum_{i \in N} \cdot(\alpha_i \cdot \text{HPWL}_i) \tag{7}$$

while satisfying the delay constraint shown in Inequality (6). We use the following formula for δ_i:

$$\begin{aligned} \delta_i = & \quad 50(\text{um}) \quad \text{if } \text{HPWL}_i \leq 1(\text{um}) \\ = & \quad 0.05 \cdot \text{HPWL}_i + 49.95(\text{um}) \quad \text{if } 1(\text{um}) \leq \text{HPWL}_i \end{aligned}$$

by which we allow very short nets (shorter than 1um) to have an almost 50um length margin and all the other nets to have a margin of (49.95um + 5% of the length of the net in the 2D layout). δ_i is fine-tuned for a given process technology by exhaustive delay simulation.

B. Finding Optimal Locations

For each cell in a given 2D placement result, we find an optimal location that can minimize the sum of the dynamic power of all the nets connected to the cell. The idea is to move a cell in a direction we can reduce the sum of the dynamic power. For example, suppose Cell A in Figure 3 is connected to Net 1 and Net 2. If α_1 is greater than α_2, the optimal location for Cell A is $(x_1, [y_5, y_4])$. If α_2 is greater than α_1, however, the optimal location for Cell A is $(x_5, [y_5, y_4])$.

The following theorem shows how to find optimal locations that minimize the dynamic power consumption for each cell.

Theorem 1: For Cell A connected to k nets (n_1, ..., n_k), we construct two bounding boxes, one ($B_{q,1}$) without Cell A and the other ($B_{q,2}$) with Cell A, for n_q ($1 \leq q \leq k$). Let B_A be a set of all those bounding boxes, $B_A = \{B_{1,1}, B_{1,2}, B_{2,1}, ..., B_{k,2}\}$ and EP_A be a set of all extremal points (four end points) of all the bounding boxes in B_A. Let T_A be a set of all intersection points of all pairs of the bounding boxes in B_A. Then, 1) the current location of Cell A is optimal or 2) there exists at least one optimal point in $T_A \cup EP_A$ that minimizes the sum of the dynamic power of all the nets connected to Cell A.

Proof: The objective function we minimize is $\lambda = \sum_{i \in N} \cdot(\alpha_i \cdot \text{HPWL}_i)$ where HPWL_i is computed by $|x_{\max} - x_{\min}| + |y_{\max} - y_{\min}|$. λ is piecewise linear, so optimal points exist in some closed rectangles or closed intervals (lines), or on some extremal points (endpoints of some intervals or boundary lines of some rectangles). Since the closed rectangles and intervals include their endpoints, at least one of the extremal points or the intersection points are optimal. ■

We minimize dynamic power consumption by finding an optimal location for each cell and moving the cell to the location using the above theorem. However, we should consider the delay and density constraints because the optimal locations found by the above theorem could violate the delay constraint and/or move too many cells into a small area. We explain how we consider the constraints in the next section.

C. Delay and Density Constraints

Optimal locations of some cells found by Theorem 1 might violate the delay constraint in Inequality (6). Thus, before we move a cell to its optimal location, we check whether the move will violate the delay constraint. If it violates the constraint, we move the cell to the farthest location from the current location, satisfying the delay constraint, along and inside the segment connecting the current and the optimal locations of the cell.

As mentioned in the previous section, moving cells to their optimal locations might increase the density of a layout area significantly. Thus we need to control the density efficiently. In this work, we pre-determine a bin size, analyze the given 2D layout, obtain the maximum density in the layout, and limit the density of each bin to be at most the maximum density. If the density of the bin where the optimal location resides is already violating the density constraint, we move the cell to the next closest bin satisfying our density constraint.

We satisfy the timing and density constraints for each move by considering both at the same time. Thus, we guarantee that we never violate the timing and density constraints during/after moving cells.

D. Clustering

A problem we found in moving a cell individually to its optimal location is that high-activity nets dominate the dynamic power consumption of the cells connected to the nets. For example, moving Cell A, Cell B, and Cell C toward Cell D in Figure 4 will reduce the dynamic power consumption, but moving the three cells one by one is prohibited because moving each one of them increases the dynamic power consumption. Thus, we also cluster the cells connected to high-activity nets and move the cells simultaneously to reduce dynamic power consumption further.

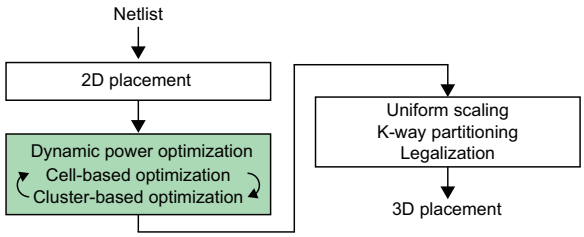

Fig. 5. Our 3D IC design flow. The uniform-scaling-based placement skips the dynamic power optimization step.

V. Simulation Results

In this section, we present our simulation results and detailed analysis.

TABLE III

Comparison of 2D, k-tier uniform-scaling-based (kTU), and k-tier non-uniform-scaling-based (kTNU) placement results. The values in parentheses show the ratio to the 2D results. FP is the footprint area.

Benchmark	Design	FP(um^2)	HPWL(m)	Power (α·HPWL)
LDPC	2D	600x600	4.58 (1.000)	3.12 (1.000)
	2TU	429x429	3.44 (0.750)	2.25 (0.722)
	3TU	353x353	3.03 (0.662)	1.89 (0.606)
	4TU	307x307	2.80 (0.611)	1.68 (0.539)
	2TNU	429x429	3.17 (0.692)	1.97 (**0.631**)
	3TNU	353x353	2.81 (0.613)	1.65 (**0.529**)
	4TNU	307x307	2.61 (0.569)	1.48 (**0.474**)
DES	2D	529x527	0.98 (1.000)	0.15 (1.000)
	2TU	379x377	0.72 (0.738)	0.11 (0.737)
	3TU	312x310	0.61 (0.623)	0.096 (0.622)
	4TU	271x270	0.54 (0.555)	0.086 (0.554)
	2TNU	379x377	0.66 (0.677)	0.099 (**0.639**)
	3TNU	312x310	0.56 (0.575)	0.084 (**0.545**)
	4TNU	271x270	0.50 (0.515)	0.076 (**0.488**)
FFT	2D	1058x1050	6.07 (1.000)	1.18 (1.000)
	2TU	753x747	4.32 (0.711)	0.85 (0.718)
	3TU	617x612	3.55 (0.585)	0.70 (0.593)
	4TU	536x532	3.09 (0.510)	0.62 (0.519)
	2TNU	753x747	4.16 (0.685)	0.72 (**0.610**)
	3TNU	617x612	3.43 (0.565)	0.59 (**0.504**)
	4TNU	536x532	3.01 (0.496)	0.52 (**0.445**)
M256	2D	767x766	9.60 (1.000)	1.90 (1.000)
	2TU	547x547	7.05 (0.734)	1.50 (0.789)
	3TU	449x448	5.93 (0.617)	1.33 (0.697)
	4TU	390x390	5.26 (0.548)	1.22 (0.642)
	2TNU	547x547	6.86 (0.714)	1.42 (**0.748**)
	3TNU	449x448	5.78 (0.602)	1.26 (**0.664**)
	4TNU	390x390	5.13 (0.535)	1.17 (**0.614**)

A. 3D IC Design Flow and Simulation Setup

Figure 5 shows the overall design flow for USBP and NUSBP. USBP skips the dynamic power optimization step. For NUSBP, we iterate the cell-based optimization and cluster-based optimization multiple times until the power reduction saturates in the dynamic power optimization step. We use the Nangate 45nm library [9] for the standard cell library, Synopsys Design Compiler for synthesis, and Cadence Encounter for 2D placement and legalization. We use hMetis [10] for the k-way partitioning to design k-tier monolithic 3D ICs. We sequentially apply hMetis to each partitioning bin of size $5 * r$ by $5 * r$ where r is the height of a standard cell row for balanced partitioning. The bin size for the density check is 50um by 50um. All the results of NUSBP do not violate the delay and density constraints.

B. Comparison of Dynamic Power Consumption in Two-Tier Monolithic 3D ICs

Table III shows wirelength ($\sum$HPWL) and dynamic power consumption ($\sum \alpha$·HPWL) of 2D and two-tier monolithic 3D ICs designed by USBP (denoted by 2TU) and NUSBP (denoted by 2TNU). As the table shows, the USBP algorithm reduces the dynamic power consumption by 22% to 28% compared to the 2D placement algorithm and our NUSBP algorithm reduces the dynamic power consumption by 25% to 39% compared to the 2D placement algorithm. In addition, the NUSBP algorithm constantly outperforms the USBP algorithm for all the benchmarks by 5% to 15%.

For more detailed analysis, we show the difference between the power consumption of 2TNU and 2TU for each net in Figure 6. In Figure 6(a), we group all nets into each switching activity bin of width 0.001, compute the sum of the dynamic power of the nets in each bin for 2TNU and 2TU, and plot their difference. In the figure, we observe that the total power reduction comes primarily from the power reduction in high-activity nets. For instance, the difference between the sum of the dynamic power consumption of high-activity nets ($\alpha \approx 1.0$) in 2TNU and that in 2TU is almost -36,000. Similarly, 2TNU reduces the dynamic power consumption of high-activity nets ($\alpha \geq 0.8$) further. However, some low-activity nets ($\alpha \leq 0.4$) in 2TNU have higher power than those in 2TU. This is unavoidable because the further power reduction in NUSBP is due to making high-activity nets shorter and low-activity nets longer.

In Figure 6(b), we group all nets into each switching activity bin of interval width 0.001, compute the sum of the HPWL of the nets in each bin for 2TNU and 2TU, and plot their difference. In the figure, we observe that the low-activity nets of switching activity around 0.2 are shortened significantly, but their contribution to the total power reduction is small due to their low activity. On the other hand, the high-activity nets of switching activity around 1.0 are shortened by half of the wirelength reduction of the low-activity nets, but their contribution to the total power reduction is high. Figure 7 shows zoom-in shots of Figure 6.

C. Comparison of Dynamic Power Consumption in Multi-Tier Monolithic 3D ICs

Table III shows that the multi-tier monolithic 3D ICs reduce the dynamic power consumption more effectively than the two-tier monolithic 3D ICs. The USBP algorithm outperforms the 2D layout by 22% to 28%, 31% to 41%, and 36% to 48% by two-, three-, and four-tier designs, respectively. In addition, our NUSBP algorithm outperforms the USBP algorithm by 5% to 15%, 5% to 15%, and 4% to 16% for two-, three-, and four-tier designs, respectively. Although the dynamic power consumption monotonically decreases as the tier count goes up, the decrement also reduces. Thus, the dynamic power reduction will eventually saturate even if more tiers are stacked. This is due to the saturation in the

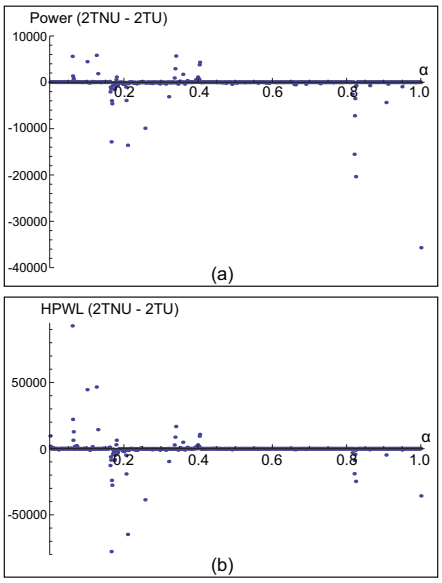

Fig. 6. Comparison of (a) power and (b) HPWL between 2TNU and 2TU. The x-axis is the net activity. Benchmark: FFT.

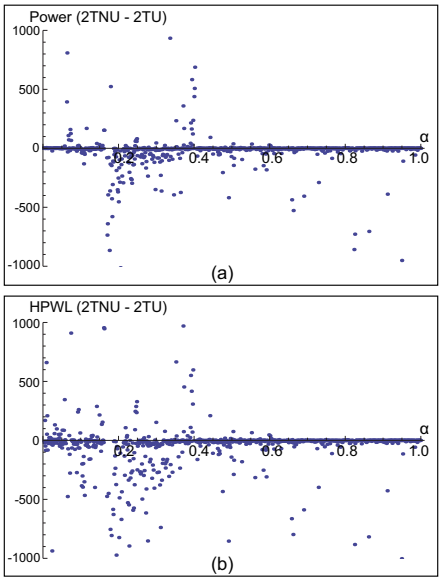

Fig. 7. Zoom-in shots of Figure 6.

wirelength reduction as shown in the same table. Since the amount of wirelength reduction is proportional to the scaling ratio ($1/\sqrt{N_{\mathrm{T}}}$), wirelength reduction saturates, which is also translated into the saturation of the dynamic power reduction. However, our NUSBP algorithm still outperforms the USBP algorithm constantly in the multi-tier monolithic 3D designs.

D. Complexity Analysis

The cell-based power optimization finds an optimal location for each cell and moves the cell to the location. Suppose a target cell is connected to maximum n nets, each of which connects maximum c cells. Then, the complexity of finding two bounding boxes (one with the cell included and the other without the cell) for each net is $\mathcal{O}(c)$ and that of finding all bounding boxes of the target cell is $\mathcal{O}(cn)$. Then, finding intersection points of the bounding boxes takes $\mathcal{O}((cn)^2)$, so the complexity of finding an optimal location for each target cell is $\mathcal{O}((cn)^2)$. Practically, n and c are bounded, so the complexity of finding an optimal location for a cell is $\mathcal{O}(1)$. For an optimal location found by the above process, finding an optimal location that satisfies the delay and density constraints takes a constant amount of time, so the complexity of moving all the cells to their optimal locations is $\mathcal{O}(C)$ where C is the total number of cells. The cluster-based optimization also moves a cluster for each net, so practically the complexity of finding an optimal location for a cluster is also $\mathcal{O}(1)$. Thus, the complexity of moving all the clusters to their optimal locations is $\mathcal{O}(N)$ where N is the total number of nets. We iterate the cell-based and cluster-based optimizations only a few times, so the overall complexity of our NUSBP algorithm is practically $\mathcal{O}(N + C)$.

VI. Conclusion

In this paper, we proposed a non-uniform-scaling-based detailed placement algorithm (NUSBP) for dynamic power optimization in multi-tier gate-level monolithic 3D ICs. Our algorithm finds an optimal location minimizing the dynamic power consumption of the sum of the nets connected to the cell for each cell without violating the delay and density constraints. The simulation results show that our algorithm outperforms the uniform-scaling-based placement algorithm by 5% to 15%, 5% to 15%, and 4% to 16% for two-, three-, and four-tier monolithic 3D ICs.

References

[1] C.-H. Shen, J.-M. Shieh, T.-T. Wu, W.-H. Huang, C.-C. Yang, *et al.*, "Monolithic 3D Chip Integrated with 500ns NVM, 3ps Logic Circuits and SRAM," in *Proc. IEEE Int. Electron Devices Meeting*, Dec. 2013, pp. 9.3.1–9.3.4.

[2] D. Henry, X. Bailin, V. Lapras, M. Vaudaine, J. Quemper, *et al.*, "Via First Technology Development Based on High Aspect Ratio Trenches Filled with Doped Polysilicon," in *IEEE Electronic Components and Technology Conf.*, May 2007, pp. 830–835.

[3] J. U. Knickerbocker, P. S. Andry, B. Dang, R. R. Horton, M. J. Interrante, *et al.*, "Three-Dimensional Silicon Integration," in *IBM Journal of Research and Development*, Nov. 2008, pp. 553–569.

[4] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Design and CAD Methodologies for Low Power Gate-level Monolithic 3D ICs," in *Proc. Int. Symp. on Low Power Electronics and Design*, Aug. 2014, pp. 171–176.

[5] D. H. Kim, S. Mukhopadhyay, and S. K. Lim, "TSV-Aware Interconnect Distribution Models for Prediction of Delay and Power Consumption of 3-D Stacked ICs," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 9, Sept. 2014, pp. 1384–1395.

[6] S. Bobba, A. Chakraborty, O. Thomas, P. Batude, T. Ernst, *et al.*, "CELONCEL: Effective Design Technique for 3-D Monolithic Integration Targeting High Performance Integrated Circuits," in *Proc. Asia and South Pacific Design Automation Conf.*, Jan. 2011, pp. 336–343.

[7] Y.-J. Lee and S. K. Lim, "Ultrahigh Density Logic Designs Using Monolithic 3-D Integration," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 12, Dec. 2013, pp. 1892–1905.

[8] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "High-Density Integration of Functional Modules Using Monolithic 3D-IC Technology," in *Proc. Asia and South Pacific Design Automation Conf.*, Jan. 2013, pp. 681–686.

[9] Nangate, "Nangate 45nm Open Cell Library," http://www.nangate.com.

[10] G. Karypis and V. Kumar, "hMETIS, a Hypergraph Partitioning Package Version 1.5.3," http://glaros.dte.umn.edu/gkhome/metis/hmetis/download.